Audio Engineering Society

Convention Paper 10198

Presented at the 146th Convention
2019 March 20 – 23, Dublin, Ireland

Tagging and retrieval of room impulse responses using semantic word vectors and perceptual measures of reverberation

Emmanouil Theofanis Chourdakis[1] and Joshua D. Reiss[1]

[1]*Queen Mary University of London*

Correspondence should be addressed to Emmanouil Theofanis Chourdakis (e.t.chourdakis@qmul.ac.uk)

ABSTRACT

This paper studies tagging and retrieval of room impulse responses from a labelled library. A similarity-based method is introduced that relies on perceptually relevant characteristics of reverberation. This method is developed using a publicly available dataset of algorithmic reverberation settings. Semantic word vectors are introduced to exploit semantic correlation among tags and allow for unseen words to be used for retrieval. Average precision is reported on a subset of the dataset as well as tagging of recorded room impulse responses. The developed approach manages to assign downloaded room impulse responses to tags that match their short descriptions. Furthermore, introducing semantic word vectors allows it to perform well even when large portions of the training data have been replaced by synonyms.

1 Introduction

Artificial reverberation has been used extensively in the process of mixing music by sound engineers to introduce the illusion of space and make instruments and voice sound more natural. There are two distinct ways to produce reverberation with computational means: by imitating the reverberation process using algorithmic approaches, or by convolving with a *room impulse response* (RIR) which tries to capture the behaviour of a space across time and frequency [1]. While algorithmic approaches predate convolutional historically, the latter became prevalent in cases when sound engineers wanted to assign a label (e.g. forest, underground park) to the acoustics of an environment. Those approaches rely on using a convolutional reverberation plug-in in the sound engineer's digital audio workstation and a library of RIRs. These RIRs are stored as audio files and up to now needed be manually annotated (usually in their filename), and retrieving them relied on simply searching the text for these annotations. This may raise issues when the files are labelled incorrectly or not according to their perceived characteristics (e.g. `ir.wav` instead of `forest.wav`). In this paper we present a way to alleviate those issues, by automatically tagging and retrieving unlabelled RIRs based on the perceptual effect they have on sound. For example we can search RIRs that make a sound *loud* and *dark* based on their content. Our main motivation for this is our previous work on producing sound from story narrative [2] where we needed methods to apply effects based on the source story text. We believe that our method can lead to assistive tools for sound engineers, allowing them to browse a library of RIRs easily; aid in field recording

scenarios [3] by quickly organising recorded RIRs; and organising and retrieving the numerous freely available RIRs available on the net.

Since RIRs are stored as regular audio files, an approach would be to use content-based audio retrieval tools to label and retrieve them [4]. Those approaches however make only minor assumptions about the features of the sound, and try to learn the parts that are useful in retrieval by analysing each separate frame of the signal and using sophisticated machine learning tools to assign labels to audio files. However, RIRs have been studied extensively in the past and their perceptually relevan characteristics are well known. We exploit these characteristics to provide a retrieval system for RIRs. Our main contributions in this paper are:

1. We provide a similarity-based method for tagging and retrieving RIR files based only on their perceptually relevant characteristics.
2. We show that by carefully choosing representation of words, we can retrieve RIRs by querying for labels we have not seen before.
3. We show that with the correct choice of RIR characteristics such a method can lead to models that are trained on datasets of synthetic RIRs, but can also be used for recorded RIRs.

2 Previous Work

In [5] the authors described an algorithmic reverberation effect that can be controlled by perceptually relevant measurements of the reverberation impulse response, such as reverberation time and echo density, to apply reverberation based on specific terms (e.g. *boomy* or *not boomy at all*). The work continued in [6] which created a map from those terms and applied reverberation either by searching for a specific term, or by exploring the descriptor map. The authors of [7] presented an effect plugin architecture for algorithmic reverberation that allows crowdsourcing of semantic descriptors from the users of the effect. Our paper is similar to the works above in that it tries to apply reverberation using crowd-sourced semantic descriptors but differs in that it allows applying reverb using multiple descriptors (for example *dark* and *muffled* instead of just *dark* or *muffled*), and it does so using convolutional reverberation and recorded room impulse responses instead of an algorithmic reverb effect.

3 Retrieval based on similarity

We approach the problem of retrieving RIRs from text queries in a similar fashion to [4]. A content-based retrieval system that can retrieve RIRs from queries has the goal of taking a set of RIRs M, and a query q and ranking them so that a RIR that is more relevant to q than m' gets ranked earlier:

$$r(m,q) < r(m',q) \tag{1}$$

Similarly, a system that assigns tags t in T to RIRs m in M should rank tag t that is more relevant to m than t' ahead of it:

$$r(m,t) < r(m,t') \tag{2}$$

In order to construct such a system, we can construct functions F such that:

$$F(m,q) > F(m',q) \tag{3}$$
$$F(m,t) > F(m,t') \tag{4}$$

A simple but effective method is to count occurrences of pairs (m,t) in a training set, where m is an impulse response and t a tag, and use those occurrences in a matrix as our scores (normalized so that each row has length 1 for convenience). Suppose we have a query q consisting of tags $t_1 \dots t_N$ and a matrix of occurrences $\boldsymbol{W}$, we can define the score as:

$$F(m,q) = \boldsymbol{W}_{m,t_1} + \boldsymbol{W}_{m,t_2} + \cdots + \boldsymbol{W}_{m,t_N} \tag{5}$$

If we represent tags as a set of column vectors $\boldsymbol{t} \in [0,1]^N$, we can write the above equation as:

$$F(m,q) = \boldsymbol{w}_m \boldsymbol{q} = < \boldsymbol{w}_m^T, \boldsymbol{q} > \tag{6}$$

where $\boldsymbol{w}_m$ is the row of the occurrence matrix $\boldsymbol{W}$ corresponding to impulse response m, $< \cdot, \cdot >$ represents the vector inner product, and $\boldsymbol{q}$ is the sum of tags $\boldsymbol{t}_{1\dots N}$. It is worth noting that the inner product is a measure of *similarity*, the more similar $\boldsymbol{w}^{\boldsymbol{T}}$ is to $\boldsymbol{t}$, the higher the value of the product. By using the vector identify for the inner product:

$$< \boldsymbol{a}, \boldsymbol{b} > = \|\boldsymbol{a}\| \, \|\boldsymbol{b}\| \cos \angle(\boldsymbol{a}, \boldsymbol{b}) \tag{7}$$

We can further write Eq. 6 as:

$$F(m,q) = < \boldsymbol{w}_m^T, \boldsymbol{q} > \quad (8)$$
$$= \left\|\boldsymbol{w}_m^T\right\| \|\boldsymbol{q}\| \cos\angle(\boldsymbol{w}_m^T, \boldsymbol{q}) \quad (9)$$
$$= \cos\angle(\boldsymbol{w}_m^T, \boldsymbol{q}) \quad (10)$$

Where $\|\cdot\|$ is the euclidean vector norm, and the quantity $\cos\angle(\boldsymbol{w}^T, \boldsymbol{t})$ is the *cosine similarity* between $\boldsymbol{w}^T$ and $\boldsymbol{t}$. We can constrain $\|\boldsymbol{q}\| = 1$ (e.g. by dividing it by its length) and we can have also constrained vector $\boldsymbol{w}^T$ to have length 1.

In order to find the k most relevant RIRs m to query q, we (1) calculate the cosine similarities between $\boldsymbol{q}$ and all $\boldsymbol{W}$, (2) sort them in descending order, and (3) we select the first k. A block diagram of the process can be seen in Fig. 1.

To find the k most relevant tags for a specific RIR m we work in a similar fashion. However instead of cosine similarity on the occurrence matrix $\boldsymbol{W}$, we (1) check euclidean distances between m, characterised by a feature vector d_m that characterizes the RIR, and all of the RIRs λ in our dictionary, characterised by feature vectors d_λ, (2) sort them in ascending order, and (3) pick the first k tags that correspond to the top labels in that step. A block diagram of the process can be seen in Fig. 2. This depends on the assumption that similar RIRs are going to be labelled similarly, which we have found works in practice. The choice of feature vectors d_m is explained in section 5.

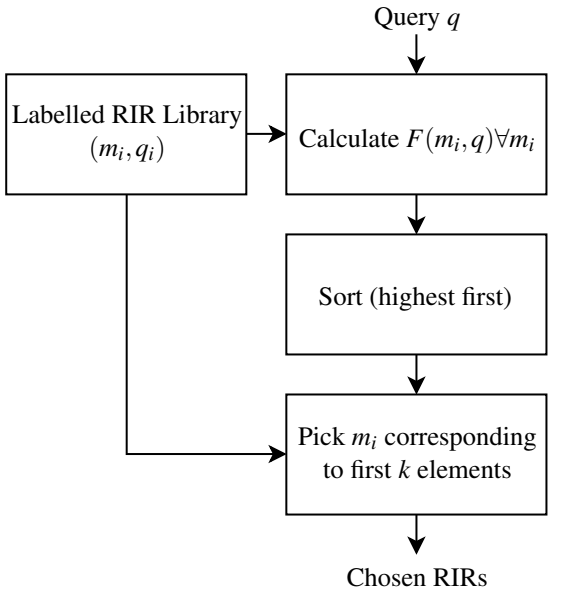


Fig. 1: Block diagram for retrieving the k most relevant RIRs to query q.

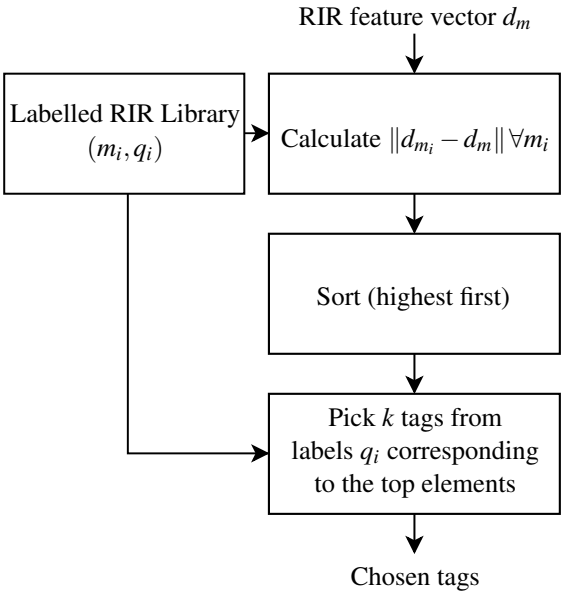


Fig. 2: Block diagram for retrieving the k most relevant tags to RIR m.

4 Choice of dictionary for Query-to-RIR retrieval

The methods described above work for every dictionary of tags chosen where $\boldsymbol{t} \in [0,1]^N$. One obvious choice is for each tag n to map to Kronecker's delta:

$$\boldsymbol{t}_n = \delta_n \quad (11)$$
$$\boldsymbol{q} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{t}_i \quad (12)$$

Kronecker's delta is given by:

$$\delta_n = \begin{cases} 1 & \text{if tag is the n-th tag} \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

Eq. 12 comes from the constraint $\|q\| = 1$ in case the query consists of M tags. The method we described in section 3 then works when every retrieval request contains a subset of the tags it has been trained with. This choice of dictionary however, has two major limitations; it does not take into account semantic relationships between different words (e.g. between *big* and *large*), and it cannot deal with out-of-dictionary words.

The second case is obvious. If N words in the dictionary are represented with $\delta_{1...N}$, then an out-of-dictionary word will be represented as δ_ν, with $\nu > N$ which is of higher dimensionality than our dictionary. It is less obvious why the first case poses an issue, but suppose we have two RIRs m_1 and m_2, labelled as *big-hall*, and *large-hall* respectively, and we query our retrieval system for RIRs that match *big-hall*. We would like our system to return to us both m_1 and m_2 since they are relevant. However, *big-hall* and *large-hall* are labelled with different δ_ν vectors which are by definition orthogonal and therefore their cosine similarity is 0. So if we query for *big-hall* , m_2 which is labelled as *large-hall* will not be retrieved (and vice versa).

These limitations can be addressed by using a different set of vectors $\boldsymbol{t}_\nu$ for our dictionary, which retain semantic relationships between the dictionary terms, as long as those vectors satisfy $\|\boldsymbol{t}_\nu\| = 1$. An appropriate set of such vectors can be found in the form of semantic vectors (or word embeddings). They assign a high dimensional vector to each distinct word such that the distance between two different words relate to some kind of semantic function (such as encoding whether the two words are regularly used together). Example of such word embeddings include Word2Vec [8], GloVe [9], and ConceptNet Numberbatch [10]. In this work we make use of the last one, since it encodes a type of commonsense similarity. For example, a *small-room* will be assigned closer to a *chapel* than a *cathedral*.

An advantage of using word embeddings in our dictionary is that a similarity between different words is already embedded in their vector representation v_{word}, which can be used simply by replacing the orthonormal basis function give in Eqs. 11, and 12 with:

$$\boldsymbol{t}_n = v_{word} \quad (14)$$

$$\boldsymbol{q} = \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{t}_i \quad (15)$$

This way, when we query for *big-hall*, the cosine similarity with $\boldsymbol{w}_{m_2}$ above will be high enough that we retrieve m_2 and therefore solve the first issue described above. Additionally, out-of-dictionary terms stop being a problem since every new word will have the same dimensionality as every word in our dictionary and therefore can be compared against them using cosine similarity.

In the section 6, we present experimental results and show one method based on a similarity method referred to as *sim*, using the simple orthonormal basis set, as well as one using the numberbatch embeddings instead, which we will refer to as *cb*.

5 Acoustic Features

In section 3, we mentioned that a RIR m is characterised by a feature vector $\boldsymbol{d}_m$ and that we use similarity between a RIR m and each RIR λ in our library to retrieve the most relevant tags based on the labels of the RIRs that are most similar to m. In this section we explain what the features in $\boldsymbol{d}_m$ are and how they are derived.

Since our RIRs are essentially audio recordings we could use methods for content-based sound retrieval similar to the ones presented in [4]. For example, by extracting the frames of the audio signal, fitting a Gaussian Mixture Model for each tag, and using the average of the log-likelihood of each model and each frame as our scoring function F to rank each RIR. This would require extracting and fitting a model for at least hundreds of frames for each impulse response.

Compared to arbitrary audio files however, RIRs have been studied extensively and a number of perceptual characteristics can be extracted that can sufficiently describe them. Instead of extracting hundreds of frames for each recording we can therefore just extract a handful of those characteristics. There are a lot of those features to choose from. For this work we chose the following:

- Reverberation time T_{60} [11].

- Echo Density E_d [12].
- Direct-to-Reveberant Ratio D_{rr} [13].
- Central Time T_c.
- Spectral Centroid S_c.

to construct the vector $\boldsymbol{d}$ which characterises each RIR:

$$\boldsymbol{d} = [T_{60}\ E_d\ D_{rr}\ T_c\ S_c]^T \tag{16}$$

We chose those measurements over others because they can be directly mapped to the features used in [6] and can be computed equivalently for both algorithmically synthesised and recorded RIRs. They therefore allow us to use the dataset provided in [6, 14] with minimal effort. Below we give a definition for each characteristic and a summary on how it is derived.

5.1 Reverberation Time T_{60}

Reverberation time is the time it takes for the power of the impulse response to drop to imperceptible levels, usually below $60dB$ for T_{60}. For RIRs synthesised by algorithmic reverberation, this is usually just a matter of measuring the levels of the impulse response dropping below a relative 10^{-3} compared to the direct sound [5]. For recorded RIRs however, the presence of noise makes this way of measuring T_{60} unreliable since the noise level will probably be higher than that. In order to measure reverberation time at $60db$ therefore, we use PYTHON-ACOUSTICS[1] which implements a method based on [15]. This consists of measuring the slope of the RIR at an earlier threshold (in our case $30dB$) and interpolating to $60dB$. Furthermore, T_{60} is not measured at the full spectrum, but at 8 octaves centred at $C_f = \{63, 125, 250, 500, 1000, 2000, 4000, 8000\}Hz$ and the average is taken as the chosen reverberation time T_{60}:

$$T_{60} = \frac{1}{8} \sum_{f \in C_f} T_{60}^f \tag{17}$$

where each superscript f is the centre frequency of the respective band.

[1] https://github.com/python-acoustics

5.2 Echo Density E_d

Echo density is the number of distinct reflections of the direct sound that can be heard. It is usually measured from the impulse response during the early phase of the reverberation. Again, as with reverberation time, echo density can be directly measured in the case of an algorithmically synthesised RIR by measuring the number of echoes (peaks) over a small period of time at the early reverberation phase. Again, as with reverberation time it is not as easy to measure in the case of recorded RIRs. To do that, we use the method described in [12] which does not give the echo density exactly, but a metric that highly correlates with it. In our case, we measure this metric at 8 different time instances $T = \{5, 10, 15, 20, 30, 50, 90, 190\}ms$ on the RIR and take the average.

$$E_d = \frac{1}{8} \sum_{\tau \in T} E_d^\tau \tag{18}$$

5.3 Direct-to-Reverberant Ratio D_{rr}

In [5, 6, 14], the authors base their work on a measure called Clarity C. However, by the way it is defined it is more relevant to the Direct-to-Reverberant ratio D_{rr} when measured in recorded RIRs. This describes the ratio of energies of the direct sound, to the rest of the reverberation, expressed in dB.

$$D_{rr} = \frac{E_{direct}}{E_{reverberant}} \tag{19}$$

In the algorithmically synthesised case, it is a simple matter of taking the ratio of the RIR value at time $t = 0$ to the sum of squares of the rest of the values. In the recorded RIR case again this is not the case since we are uncertain of the location and duration of the direct sound. To estimate D_{rr} we use the method described in [13] which calculates the ratio of energies around a $5ms$ window around the highest peak of the signal and divides it by the rest of the RIR signal.

$$D_{rr} = \frac{\int_{t_0-2.5ms}^{t_0+2.5ms} y^2(\tau)d\tau}{\int_{t_0+2.5ms}^{T} y^2(\tau)d\tau} \tag{20}$$

where $y(t)$ is the signal of the RIR, t_0 the location of the highest peak, and T the total duration of the impulse response.

5.4 Central Time T_c

Central time T_c is the centre of gravity of the impulse response [16] and in previous works has been found to be associated with perceptual descriptors such as *boomy*, or *church-like* [5]. It can be calculated the same way for algorithmic and recorded RIRs and is given by:

$$T_c = \frac{\int_0^T \tau y(\tau) d\tau}{\int_0^T y(\tau) d\tau} \tag{21}$$

where again, y is the signal of the RIR, and T its total duration.

5.5 Spectral Centroid S_c

Similar to central time, spectral centroid is the frequency at the centre of gravity of the spectrum of the RIR and is correlated to the *brightness* of the impulse response [17, 5]. In our case it is calculated using LIBROSA [18]. The signal is split into frames, its Short Time Fourier Transform (STFT) is calculated and each frame n of the STFT is normalised. The spectral centroid of frame n is then given by:

$$S_c^n = \frac{\sum_{k=1}^{K} k \cdot STFT[n,k]}{\sum_{k=1}^{K} STFT[n,k]} \tag{22}$$

Where K is the number of bins of the STFT. We calculate the spectral centroid S_C^n for frames at $T = \{5, 10, 15, 20, 30, 50, 90, 190\} ms$ and take their average to compute the final spectral centroid:

$$S_c = \frac{1}{8} \sum_{\tau \in T} S_C^{\tau f_s} \tag{23}$$

Where f_s is the sampling rate of the recorded signal.

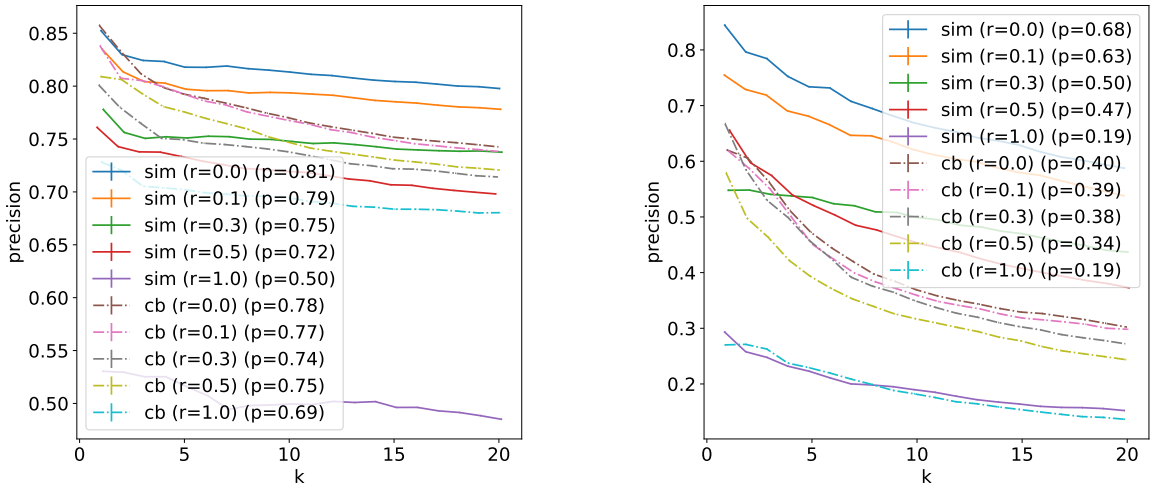
6 Experimental Results

For evaluating our tagging and retrieval method we used the dataset described in [6]. This dataset has been crowdsourced online by asking users to listen and describe, using simple words, the effect of various algorithmic reverberation settings on three excerpts of piano, guitar, and drums. To apply the effect, the algorithmic reverberator described in [5] was implemented in the browser using the Web Audio API. Since the dataset has been updated several times since its creation, we chose the version used in [14] which contained 6791 labellings of 256 different reverberation settings. While impulse response measurements were provided with the dataset, we resynthesised the impulse responses in order to extract the measurements described in section 5 which can be used both for realistic as well as synthesised impulse responses. In order to do that, we re-implemented the reverberation effect from [5] while taking into account modifications from [6] to compensate for limitations of the Web Audio API. Those modifications consisted of adding a delay of 0.1ms to the dry signal and the allpass filter, and using a biquad filter instead of the first order lowpass filter given in [5]. For this work, we report two separate results sets; precision of tagging and retrieval on a withheld part of the original dataset, and automatic tagging on 4 realistic impulse response recordings from two freely available RIR libraries [19, 20].

For the first case, we pseudo-randomly (using a predefined random seed) split our data into three equal-sized segments. We use the first two as training and development and keep the last one for testing. We report results of the methods presented in section 3 built on both the training and development set and tested on the testing set. The first two parts were used in the process of developing our models, and the last was kept completely separate in order to assure that our reported results were not biased by our development process. The models we used implement the method given in section 3 but using the two different dictionaries given in section 4. Similar to [4], we report average per-query (P_q) and per-IR (P_m) precision defined as the ratio of relevant documents retrieved at the top k positions:

$$P_\nu = \frac{|relevant_\nu \cap retrieved_\nu|}{|retrieved_\nu|}, \nu \in \{q, m\} \tag{24}$$

Where $|\cdot|$ denotes cardinality of a set. Per-query and per-IR precision curves for $k = 1 \ldots 20$ are shown in Figs. 3a, and 3b respectively. Average precision p over the curve is reported for each curve. Curve labelled as *sim* denotes precision of the similarity-based method, and *cb* the method based on the numberbatch embeddings. Ratio r is the percentage of our training labels that have been replaced with synonyms. There are cases in the dataset where impulse responses, that should be labelled the same, that were labelled using synonyms



(a) Precision curves for retrieved RIRs

(b) Precision curves for retrieved tags

Fig. 3: Precision curves for retrieval and tagging. Precision curves labelled as *sim* use an orthonormal basis for tag representation and *cb* use Numberbatch embeddings. Ratio r is the percentage of training data that have been replaced with synonyms. Average precision p reported is the percentage of relevant (a) RIRs or (b) tags retrieved respectively at the top k places.

(e.g. *big-hall* and *large-hall*) or with highly correlated words (e.g. *church*, and *cathedral*). Ignoring these correlations leads to similar impulse responses being scored (and therefore ranked) independently. In order to show why this matters we replace part of our training set each time with synonyms derived by Wordnet [21] and report their precision curves on Figs. 3a and 3b. Though the similarity-based method performs slightly better than the method based on the numberbatch embeddings, the latter loses much less in precision when provided with synonym data. This is due to the fact that the similarity based method assumes that every possible label comes from a fixed, previously known dictionary and therefore cannot deal with out-of-dictionary terms.

In table 1 we see how a system built on our method labelled four real RIRs found from OpenAIR [19] and EchoThief [20]. The first is a library of freely available impulse responses accompanied by metadata about the space and method they were recorded with, and the second is an online library of RIRs extracted from noisy environments (such as playgrounds). In the first RIR, which is from an underground car park, there is an "interesting resonance" because of some metal pipes. The system based on our method, instead of tagging with just *big* and *spacey* that relates to the car park's size, managed to tag it as *metallic*. More results on recorded RIRs on OpenAIR and EchoThief are available as supplementary material[2].

7 Discussion/Limitations

Though we showed examples of how the system performs on real recorded data in table 1, it would be desirable to conduct listening tests using listeners accustomed to the effect of the reverberation and preferably with experience in mixing. Such an experiment can be a MUSHRA [22] style listening test using the Web Audio Evaluation Tool [23] with careful choice of criteria for both listening subjects, as well as appropriate listening stimuli. Alternatively, napping experiments can be conducted for the construction of a sound wheel [24]. Designing and running such experiments however is not trivial and is out of the scope of this paper. We have also constrained the number of used perceptually relevant characteristics to the ones used in the dataset

[2] https://code.soundsoftware.ac.uk/projects/chourdakisreiss2019aes

Description / Filename	Space Category	Suggested Tags
[...] recording in an underground car park. Has a slightly nice resonance in it from the metal pipes on the ceiling	Chamber, Hall, Open Air	hall, big, deep, spacey, –, metallic church, heavy, organ, slow
The Spokane Woman's Club hall is a highly reflective space with bare walls, a hardwood floor and a curved ceiling [...]	Auditorium, Ballroom, Hall	sharp, spacious, distant, warm, echo strong, bright, electric, vibrant, cool
Outback Climbing Center	Recreation	spacey, big, room, buffled, echo deep, hollow, distant, rolling, soft
Steinman Hall	Venues	nice, heavy, clear, deep, romantic sad, bass, warm, melancholy, love

Table 1: Labelling of recorded RIR. The first two rows show RIRs from OpenAIR [19] and the last two show RIRs from the EchoThief [20] library.

from [14] and also characteristics relevant to the reverberation IR itself and not to the sound it is convolved with or to the effect it will have in the final mix. If such information were known, it could be "plugged in" as prior and posterior probabilities for tagging and retrieval. For example, [25] show that there is a strong relation between musical tempo and choice of echo delay times in artificial reverberation, something which we could exploit in order to weigh tags according to the source input. On the other hand, [26] show that reverb loudness and early decay time have significant impact on the perception of a mix. Tags therefore could be ranked according to the perceptual effect they have on the mix itself and not just to a single audio source. Such an extension however would require a dataset of reverberation parameters collected on multitrack mixes.

The methods we presented here could probably be applied to other effects as well. For example, [27, 14] show that the descriptor map used in [6] can be used for equalization and compression as well, and [28] show that there are statistical correlations among the various tags when used in different audio effects. We believe that our work can incorporate such findings with only minor modifications. We believe that other interesting applications can arise from our method. For example online libraries based on user contributions that automatically tag uploaded RIRs and allow users to browse and download them similar to what websites like FREESOUND[3] do for sound effects. Similarly, a semantic reverb effect similar to the one introduced in [6] could be developed, although for convolutional instead of algorithmic reverb and with support for full query search instead of single terms. On a different angle, we could use the work done for example in [29, 30] together with dereverberation and source separation techniques [31] to build intelligent remixing tools that can act on complex queries such as: "make the guitar part gloomier".

[3] http://www.freesound.org

References

[1] Välimäki, V. et al., "More than 50 years of artificial reverberation," in *60th International Conference on Dereverberation and Reverberation of Audio, Music, and Speech*, 2016.

[2] Chourdakis, E. and Reiss, J., "From my pen to your ears: automatic production of radio plays from unstructured story text," in *15th Sound and Music Computing Conference*, pp. 148–155, 2018.

[3] Canfield-Dafilou, E. K., Callery, E., and Jette, C., "A Portable Impulse Response Measurement System," in *15th Sound and Music Computing Conference*, pp. 172–176, 2018.

[4] Chechik, G. et al., "Large-scale Content-based Audio Retrieval from Text Queries," in *1st ACM International Conference on Multimedia Information Retrieval*, pp. 105–112, 2008.

[5] Rafii, Z. and Pardo, B., "Learning to Control a Reverberator Using Subjective Perceptual Descriptors." in *10th International Society for Music Information Retrieval Conference*, pp. 285–290, 2009.

[6] Seetharaman, P. and Pardo, B., "Reverbalize: a crowdsourced reverberation controller," in *22nd ACM international conference on Multimedia*, pp. 739–740, 2014.

[7] Stables, R. et al., "SAFE: A System for the Extraction and Retrieval of Semantic Audio Descriptors," in *Extended Abstracts for the late-breaking demo session of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, 2014.

[8] Mikolov, T. et al., "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[9] Pennington, J. et al., "Glove: Global vectors for word representation," in *2014 Conference on Empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[10] Speer, R. and Lowry-Duda, J., "ConceptNet at SemEval-2017 Task 2: Extending Word Embeddings with Multilingual Relational Knowledge," in *11th International Workshop on Semantic Evaluation*, pp. 85–89, 2017.

[11] Ratnam, R. et al., "Blind estimation of reverberation time," *The Journal of the Acoustical Society of America*, 114(5), pp. 2877–2892, 2003.

[12] Abel, J. S. and Huang, P., "A Simple, Robust Measure of Reverberation Echo Density," in *121st Audio Engineering Society Convention*, 2006.

[13] Zahorik, P., "Direct-to-reverberant energy ratio sensitivity," *The Journal of the Acoustical Society of America*, 112(5), pp. 2110–2117, 2002.

[14] Zheng, T., Seetharaman, P., and Pardo, B., "Socialfx: Studying a crowdsourced folksonomy of audio effects terms," in *24th ACM international conference on Multimedia*, pp. 182–186, 2016.

[15] ISO, E., "3382-1, 2009,"Acoustics—Measurement of Room Acoustic Parameters—Part 1: Performance Spaces,"," *International Organization for Standardization, Brussels, Belgium*, 2009.

[16] Adriaensen, F., "Acoustical impulse response measurement with ALIKI," in *Linux Audio Conference Proceedings*, pp. 9–14, 2006.

[17] Sethares, W. A., Morris, R. D., and Sethares, J. C., "Beat tracking of musical performances using low-level audio features," *IEEE Transactions on speech and audio processing*, 13(2), pp. 275–285, 2005.

[18] McFee, B. et al., "librosa: Audio and music signal analysis in python," in *14th Python in Science Conference*, pp. 18–25, 2015.

[19] Murphy, D. T. and Shelley, S., "OpenAIR: An Interactive Auralization Web Resource and Database," in *129th Audio Engineering Society Convention*, 2010.

[20] SuperHOAX, "EchoThief Impulse Response Library," `http://www.echothief.com/`, 2013, accessed: 2018-10-30.

[21] Miller, G. A., "WordNet: A Lexical Database for English," *Communications of the ACM*, 38(11), pp. 39–41, 1995.

[22] ITU-R, R. B., "1534-1,"Method for the subjective assessment of intermediate quality levels of coding systems (MUSHRA)"," *International Telecommunication Union*, 2003.

[23] Jillings, N. et al., "Web Audio Evaluation Tool: A browser-based listening test environment," in *12th Sound and Music Computing Conference*, 2015.

[24] Pedersen, T. H. and Zacharov, N., "The Development of a Sound Wheel for Reproduced Sound," in *138th Audio Engineering Society Convention*, 2015.

[25] Pestana, P. D., Reiss, J. D., and Barbosa, A., "User preference on artificial reverberation and delay time parameters," *Journal of the Audio Engineering Society*, 65(1/2), pp. 100–107, 2017.

[26] De Man, B., McNally, K., and Reiss, J., "Perceptual evaluation and analysis of reverberation in multitrack music production," *Journal of the Audio Engineering Society*, 65(1/2), pp. 108–116, 2017.

[27] Seetharaman, P. and Pardo, B., "Audealize: Crowdsourced audio production tools," *Journal of the Audio Engineering Society*, 64, pp. 683–695, 2016.

[28] Stables, R. et al., "Semantic Description of Timbral Transformations in Music Production," in *22nd ACM international conference on multimedia*, pp. 337–341, 2016.

[29] Chourdakis, E. T. and Reiss, J. D., "A Machine-Learning Approach to Application of Intelligent Artificial Reverberation," *Journal of the Audio Engineering Society*, 1/2, pp. 56–65, 2017.

[30] Benito, A. L. and Reiss, J. D., "Intelligent multitrack reverberation based on hinge-loss markov random fields," in *2017 AES International Conference on Semantic Audio*, 2017.

[31] Moffat, D. and Reiss, J. D., "Implementation and assessment of joint source separation and dereverberation," in *60th International Conference on Dereverberation and Reverberation of Audio, Music, and Speech*, 2016.